



Univerzitet u Sarajevu  
Prirodno-matematički fakultet  
ODSJEK ZA MATEMATIKU

A Zmaja od Bosne 33-35, 71000 Sarajevo, BiH  
T +387 33 279 874 F +387 33 649 342  
W [www.pmf.unsa.ba/matematika](http://www.pmf.unsa.ba/matematika)  
E [matematika@pmf.unsa.ba](mailto:matematika@pmf.unsa.ba)

# Programiranje I

## Auditorne vježbe

### *Petlje*

SEAD DELALIĆ  
[delalic.sead@gmail.com](mailto:delalic.sead@gmail.com)

**Zadatak 1.** Napisati program koji od korisnika traži unos cijelog broja  $n$ . Program ispisuje  $n$  **slučajnih** cijelih brojeva. Brojevi su ispisani u jednom redu i razdvojeni su zarezom. Nakon posljednjeg broja u nizu ne nalazi se zarez.

**Zadatak 2.** Napisati program koji od korisnika traži unos broja  $n$ . Korisnik zatim unosi  $n$  brojeva. Program ispisuje razliku najvećeg i najmanjeg unesenog broja.

**Zadatak 3.** Napisati program koji od korisnika traži unos brojeva, sve dok ne unese negativan broj. Program ispisuje ukupnu sumu i proizvod unesenih brojeva.

**Zadatak 4.** Napisati program koji od korisnika traži da unese prirodan broj  $n$ . Program ispisuje okvir kvadrata dimenzija  $n \times n$  sačinjen od zvjezdica. Unutrašnjost kvadrata nije ispunjena. Na slici je dat primjer za  $n = 5$ .

```
* * * * *
*           *
*           *
*           *
*           *
* * * * *
```

**Zadatak 5.** Napisati program koji za uneseni prirodan broj provjerava da li je prost ili složen.

## Slučajni brojevi

Slučajni brojevi u programskom jeziku Python generišu se pomoću modula `random`. Da bi koristili modul, neophodno je uraditi `import` pomoću naredbe `import random`.

U zadacima za studente, najčešće se koriste dvije funkcije.

Prva je `randint(a, b, s)`, koja generiše slučajan cijeli broj veći ili jednak `a` i manji ili jednak `b`. Treći parametar `s` predstavlja korak (engl. *step*), tj. vrijednost za koju se mijenjaju vrijednosti od `a` do `b` pri generisanju. Npr. `randint(1, 5, 2)` može generisati brojeve 1, 3 i 5, dakle, krenuvši od 1, možemo generisati brojeve koristeći korak 2. Pretpostavljena vrijednost za `s` je 1.

Druga funkcija koja se koristi je funkcija `random()` koja generiše slučajan realan broj od 0 do 1 (uključeno 0 i 1). Često je neophodno tu vrijednost pomjeriti na interval realnih brojeva sa većim rasponom od 1. U tom slučaju, dovoljno je generisani broj od 0 do 1 pomnožiti sa dužinom raspona. Ukoliko je raspon pomjeren od nule, na početnu vrijednost je neophodno dodati minimalnu vrijednost koju želimo generisati. Npr. ukoliko želimo realan broj od 2 do 10, vrijednost generišemo kao `random() * (10-2) + 2`. Ovo će generisati broj od 0 do 1, pomnožiti ga sa 8 (te dobiti broj između 0 i 8). Na dobijenu vrijednost dodajemo 2, je generisan slučajan broj od 2 do 10.

Poznato je da su računari uređaji koji imaju precizno definisano ponašanje i kod kojih je kompletan princip rada takav da *slučajnosti* u pravom smislu te riječi ne postoje. Međutim, kako bi svijet igara i niz drugih aplikacija bio krajnje dosadan bez slučajnosti (zamislite igranje karti, gdje računar uvijek, ali uvijek, izvuče istu kartu), razvijeni su algoritmi koji naizgled daju slučajne vrijednosti. Vrijednosti koje su samo naizgled slučajne nazivaju se *pseudoslučajnim brojevima*. U pozadini postoji matematički algoritam koji generiše brojeve na način da osoba ne može pretpostaviti koji je naredni broj koji će biti generisan.

Algoritmi koji generišu ovakve brojeve poprilično su kompleksni, te prihvataju određen ulaz pri generisanju brojeva (na osnovu ulaza određen je i rezultat algoritma). Za isti ulaz, algoritmi uvijek generišu jednake vrijednosti. Stoga, neohodno je omogućiti da se taj ulaz razlikuje pri svakom

pokretanju programa, kako dobijeni niz generisanih brojeva ne bilo moguće predvidjeti. Često se u praksi koristi vrijeme u milisekundama proteklo od 1.1.1970. godine, mada postoji i niz drugih načina za generisanje ulazne vrijednosti. Ulazna vrijednost se naziva **sjeme** (engl. *seed*). Za različitu vrijednost, generišu se različiti nizovi slučajnih brojeva, te za jednake vrijednosti dobijamo generisanje jednakih nizova.

Programeri često nailaze na pogreške (tzv. *bugove*) u kodu. Međutim, u zadacima koji uključuju generisanje slučajnih brojeva, kao i u ostalim zadacima, greške se mogu javiti samo za neke vrijednosti. Ako se brojevi generišu slučajno, onda možemo doći u situaciju da je teško ponoviti scenarij u kojem je došlo do greške, te otkrivanje greške može značajno potrajati (zamislite da se greška događa za vrijednost 1, a mi generišemo brojeve od 1 do 10 000). Stoga, programerima je omogućeno da mogu program sa generisanjem slučajnih brojeva pokrenuti više puta i uvijek dobiti jednake vrijednosti (bez da značajno mijenjaju kod i ručno postavljaju vrijednosti). To se radi pomoću funkcije `seed(a)`, koja postavlja sjeme generatoru pseudoslučajnih brojeva na vrijednost `a`. Sjeme se postavlja na samom početku programa, prije upotrebe funkcija za generisanje slučajnih brojeva (tj. prije vrijednosti koje želimo da ostanu iste pri višestrukom pokretanju programa). Tada možemo uklanjanjem/dodavanjem te linije dobiti kompletan program koji se izvršava svaki put jednako ili svaki put različito.